SCIENCE CHINA
Information Sciences

• Supplementary File •

Physics-informed learning for collective behavior prediction

Zifan SHI[1], Qizi HUANGPENG[1], Jingjie LIANG[1], Mingze QI[1,2*], Xiaojun DUAN[1] & Gang YAN[2]

[1]*College of Science, National University of Defense Technology, Changsha 410073, P.R. China*
[2]*MOE Key Laboratory of Advanced Micro-Structured Materials, and School of Physical Science and Engineering, Tongji University, Shanghai 200092, P.R. China*

Appendix A Zero-shot cross-behavior transfer on synthetic swarms

Controlled synthetic experiments provide additional evidence on behavior transfer and agent-number robustness. For the Vicsek and modified Cucker–Smale systems, zero-shot transfer results are reported in Supplementary Table A1 and Supplementary Figure A1. PI-DLinear substantially improves the Cucker–Smale-to-Vicsek transfer, reducing RMSE from 6.231 to 0.666. In the Vicsek-to-Cucker–Smale direction, PI-DLinear achieves an RMSE of 7.612, comparable to DLinear at 7.501, indicating that the transfer benefit depends on behavioral compatibility and should not be interpreted as universal zero-shot superiority.

Transfer protocol. The synthetic transfer follows exactly the protocol of the real-data experiment (main-text Subsection 3.6): the source checkpoints are those of the main experiments (training ratio 0.7, leakage-free windowing as in Section Appendix C); for each feature type (position or velocity in each coordinate dimension), the mean and standard deviation computed on the source training windows are aggregated across agents, and the target test inputs are normalized by matching feature types; predictions are de-normalized with the same statistics, and RMSE is computed over the full state (positions and velocities) in the original physical scale. No target-behavior statistics are used at any stage. Results are averaged over three independent training seeds.

Table A1 Zero-shot cross-behavior transfer on synthetic Vicsek and modified Cucker–Smale swarms. Bold values indicate the lower RMSE between the two models for each source–target pair. The diagonal is a within-behavior control under the same transfer protocol. The Vicsek → Cucker–Smale direction remains challenging, whereas the Cucker–Smale → Vicsek transfer shows a large advantage for PI-DLinear.

Model	Source	Target behavior	
		Vicsek	Cucker–Smale
PI-DLinear	Vicsek	0.345	7.612
	Cucker–Smale	**0.666**	**1.868**
DLinear	Vicsek	**0.343**	7.501
	Cucker–Smale	6.231	3.474

Appendix B Robustness to swarm size and agent ordering

Supplementary Figure B1 evaluates variable swarm sizes. When models trained with all 40 synthetic agents are tested on random subsets, performance remains stable across $k = 1$–40. When a model trained with 40 agents is evaluated at test sizes 20–100, Vicsek predictions remain close to their native-size controls. The modified Cucker–Smale system becomes more difficult beyond the training size, especially at 100 agents, reflecting the stronger collective coupling in that system.

For real-world data, Supplementary Table B1 summarizes two protocols. In the random-subset protocol, models trained on the complete swarm are evaluated on random subsets containing k agents (main-text Figure 5). In the stronger held-out-agent protocol, models are trained using only 8, 4, and 5 agents for mobbing, transit, and fish, respectively, and evaluated on disjoint agents, including test groups larger than the training group. PI-DLinear reduces RMSE by 18.0%–18.5% on mobbing, 22.3%–23.5% on transit, and 24.4%–26.3% on fish relative to DLinear under the held-out protocol.

Finally, Supplementary Table B2 assesses sensitivity to agent indexing. Applying ten random agent permutations to a trained checkpoint changes the test RMSE by no more than 9×10^{-8} on the real datasets. Retraining from scratch under five random agent orders produces no systematic degradation. These observations are consistent with the channel-shared architecture: permuting the input agent blocks permutes the output blocks without changing the prediction error.

* Corresponding author (email: qimingze17@nudt.edu.cn)

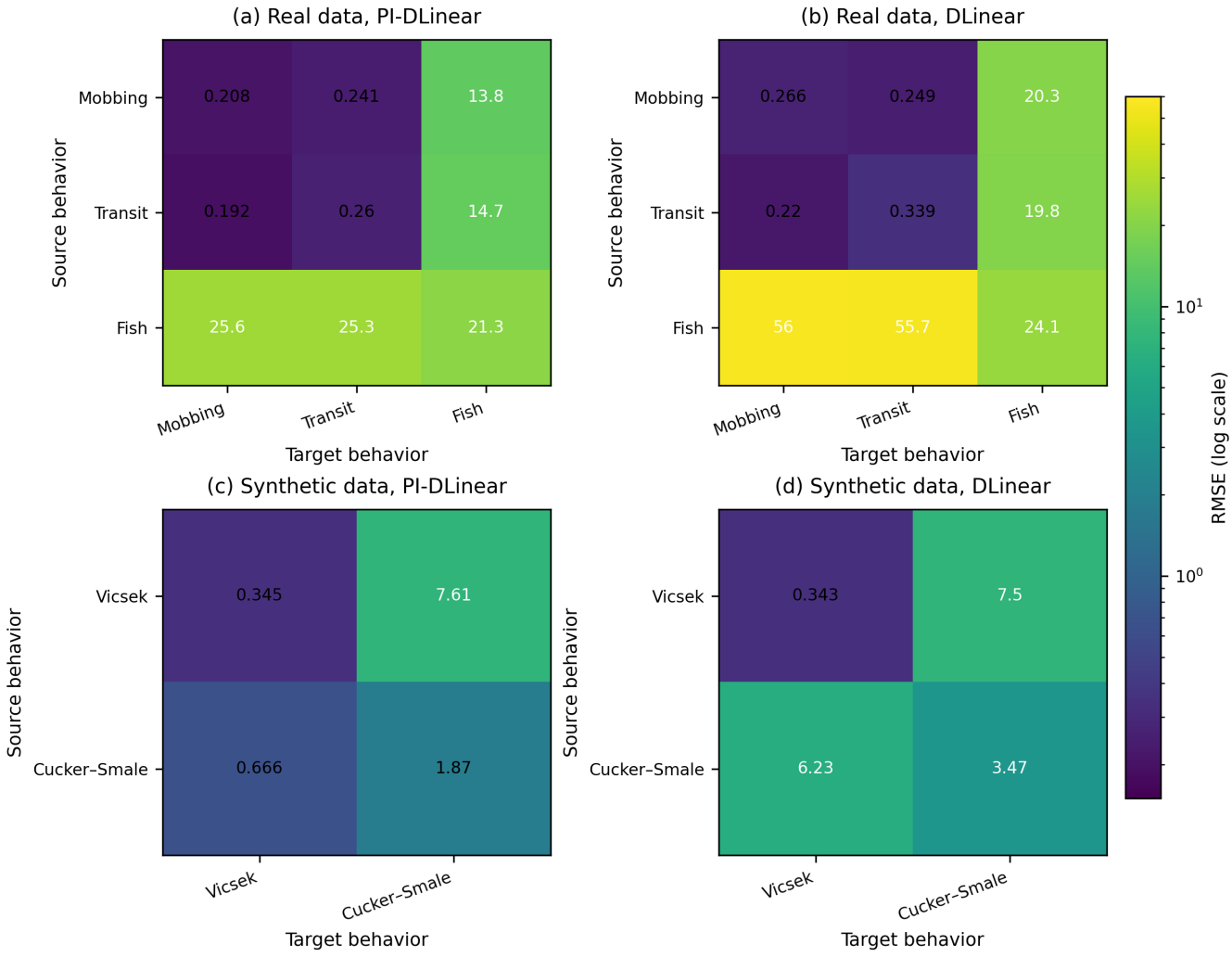

Figure A1 Zero-shot cross-behavior transfer matrices for real-world and synthetic swarms. Rows denote source behaviors and columns denote target behaviors. Color is shown on a logarithmic RMSE scale, while cell labels report raw RMSE.

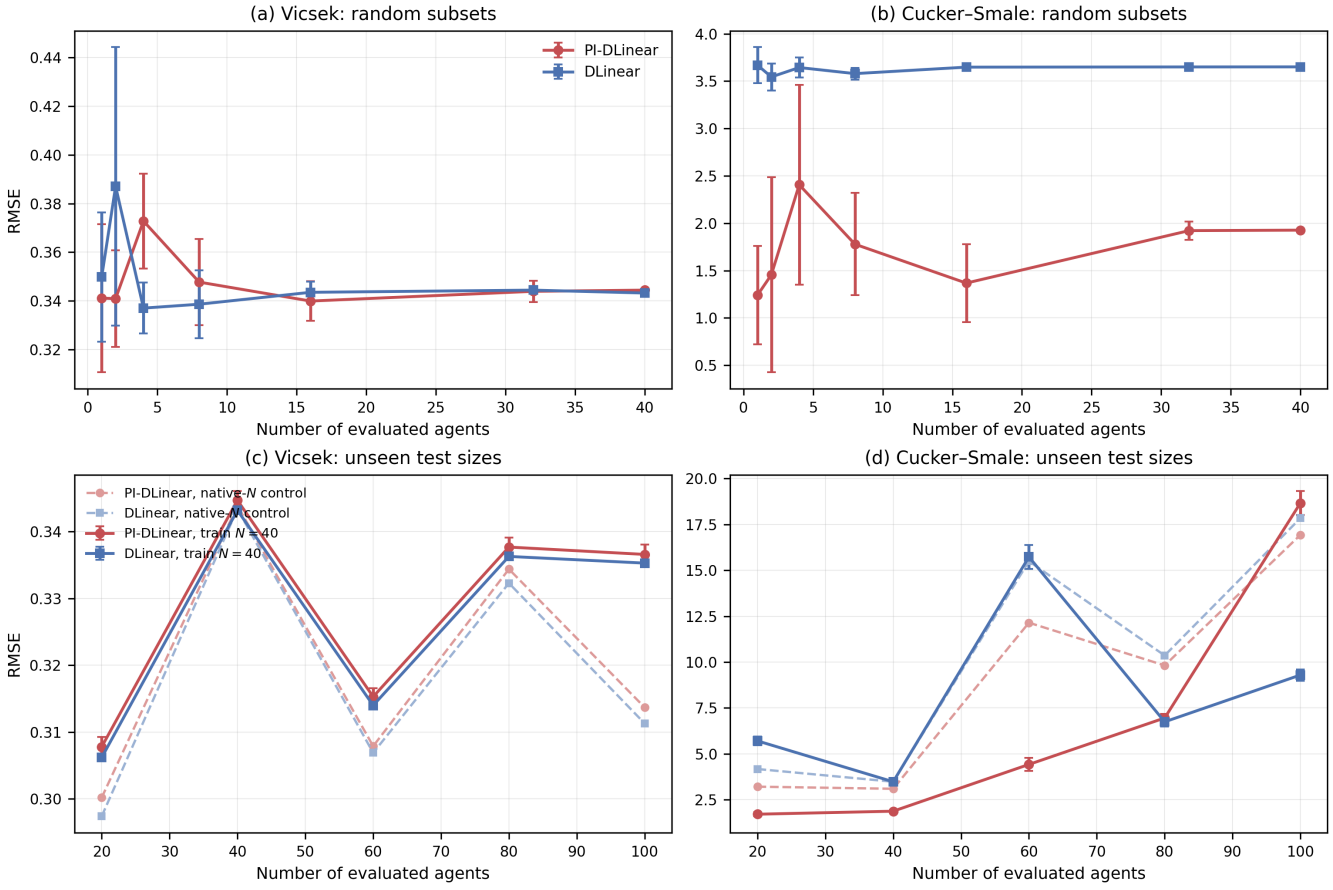

Figure B1 Swarm-size robustness on synthetic data. (a, b) Models trained with 40 agents are evaluated on random subsets. (c, d) Models trained with 40 agents are evaluated at test sizes 20–100; dashed curves show native-size controls trained separately at each test size.

Table B1 Held-out-agent and unseen-size evaluation. For real-world data, models are trained with the indicated number of agents and evaluated on disjoint held-out agents. For synthetic data, models trained with 40 agents are evaluated at the indicated test sizes; native-N controls are shown in Supplementary Figure B1. Negative reductions indicate settings where PI-DLinear is slightly above DLinear.

Data	Train N	Test N	PI-DLinear	DLinear	PI reduction
Mobbing	8	8	0.2181 ± 0.0236	0.2659 ± 0.0244	18.0%
	8	16	0.2216 ± 0.0296	0.2718 ± 0.0346	18.5%
	8	24	0.2226 ± 0.0261	0.2701 ± 0.0275	17.6%
	8	32	0.2213 ± 0.0242	0.2710 ± 0.0295	18.3%
	8	40	0.2220 ± 0.0243	0.2710 ± 0.0284	18.1%
Transit	4	2	0.2584 ± 0.0430	0.3378 ± 0.0457	23.5%
	4	4	0.2700 ± 0.0148	0.3475 ± 0.0284	22.3%
Fish	5	2	21.1405 ± 4.0685	28.6864 ± 0.8073	26.3%
	5	5	20.6291 ± 2.7977	27.2711 ± 1.0840	24.4%
Vicsek	40	20	0.3078 ± 0.0015	0.3062 ± 0.0003	−0.5%
	40	40	0.3447 ± 0.0014	0.3432 ± 0.0003	−0.4%
	40	60	0.3153 ± 0.0013	0.3140 ± 0.0002	−0.4%
	40	80	0.3377 ± 0.0014	0.3363 ± 0.0003	−0.4%
	40	100	0.3366 ± 0.0015	0.3353 ± 0.0002	−0.4%
Cucker–Smale	40	20	1.7051 ± 0.1147	5.7013 ± 0.2364	70.1%
	40	40	1.8690 ± 0.0591	3.4738 ± 0.1432	46.2%
	40	60	4.4183 ± 0.3582	15.7300 ± 0.6518	71.9%
	40	80	6.9504 ± 0.2313	6.7409 ± 0.2644	−3.1%
	40	100	18.6676 ± 0.6582	9.2926 ± 0.3104	−100.9%

Table B2 Robustness to agent-index permutation. Test-time permutation applies 10 random agent permutations to a trained checkpoint; retraining permutation retrains the model from scratch under five random agent orders.

Data	Model	Original	Test-time shuffled	Retrained mean	Retrained SD	Max. abs. diff.
Mobbing	PI-DLinear	0.235373	$0.235373 \pm 7.30 \times 10^{-9}$	0.239343	0.004210	0.009270
	DLinear	0.279119	0.279119 ± 0	0.279119	$< 10^{-7}$	$< 10^{-7}$
Transit	PI-DLinear	0.291614	0.291614 ± 0	0.280268	0.009600	0.022000
	DLinear	0.353498	$0.353498 \pm 8.94 \times 10^{-9}$	0.353498	$< 10^{-7}$	$< 10^{-7}$
Fish	PI-DLinear	20.146946	20.146946 ± 0	19.144660	0.722000	1.970000
	DLinear	27.379927	27.379927 ± 0	27.379953	3.20×10^{-5}	7.06×10^{-5}
Vicsek	PI-DLinear	0.344395	$0.344395 \pm 1.37 \times 10^{-8}$	0.344314	0.000121	0.000282
	DLinear	0.343187	$0.343187 \pm 1.49 \times 10^{-8}$	0.343187	1.46×10^{-8}	2.98×10^{-8}
Cucker–Smale	PI-DLinear	1.926716	$1.926716 \pm 5.96 \times 10^{-8}$	1.832985	0.088600	0.171000
	DLinear	3.650880	$3.650880 \pm 1.17 \times 10^{-7}$	3.650880	9.54×10^{-8}	2.38×10^{-7}

Appendix C Leakage-free chronological windowing protocol

All experiments reported in the revised manuscript use a *split-first, window-second* protocol that eliminates temporal leakage between the training, validation, and test sets.

Chronological borders. Each experiment sequence of length T is first divided along the global time axis into three disjoint segments. With training ratio r ($r \in \{0.3, 0.4, 0.5, 0.6, 0.7\}$ in the training-ratio experiments) and validation ratio 0.2, the borders are

$$T_1 = \lfloor rT \rfloor, \qquad T_2 = T_1 + \lfloor 0.2T \rfloor, \tag{C1}$$

so that the training segment is $[0, T_1)$, the validation segment is $[T_1, T_2)$, and the test segment is $[T_2, T)$.

Window construction. Sliding windows are constructed only after the split. With input length $L = 24$ and prediction horizon $H = 8$, the admissible window start indices of the three segments are

$$s \in [0,\ T_1 - L - H], \qquad s \in [T_1 + 1 - L,\ T_2 - L - H], \qquad s \in [T_2 + 1 - L,\ T - L - H], \tag{C2}$$

for training, validation, and test, respectively. A window starting at s uses the input $[s, s + L)$ and predicts the target $[s + L, s + L + H)$. Consequently, every training window — input and target — lies entirely inside the training segment; a validation (test) window may look back at most $L - 1$ steps into the preceding segment as *input history*, but all of its *predicted target steps* lie strictly inside the validation (test) segment. No window mixes target steps from two segments, so no future information from the validation or test segments can influence parameter training or checkpoint selection.

Normalization. The standardization scaler is fit on the training segment only and then applied unchanged to all segments.

Model selection and repetition. For every configuration we train up to 200 epochs with cosine learning-rate annealing, select the checkpoint with the lowest validation loss, and report the test RMSE averaged over ten independent runs with different random seeds. The LaGNA baseline follows the same chronological borders: its GNN stage is trained on the training segment, the explicit equations are fitted on the same segment, and prediction integrates the discovered equations from the final state of the validation segment across the whole test segment, after which the integrated trajectory is sliced into exactly the same $24 \to 8$ evaluation windows as all other methods.

Appendix D Baseline configurations under matched capacity and tuning budgets

Supplementary Table D1 lists the complete configuration of every compared method. All data-driven models share the same window protocol ($L = 24$, $H = 8$) and training budget: batch size 64, AdamW with initial learning rate 10^{-3}, cosine annealing over the full run, gradient clipping at 1.0, at most 200 epochs with checkpoint selection by validation loss, and ten independent runs per configuration. The hidden dimensions of the baselines follow their official implementations and were chosen so that their capacities bracket that of PI-DLinear from above; Supplementary Table D2 reports the resulting trainable parameter counts.

Table D1 Hyperparameter settings of all compared methods. Settings not listed are identical across models (window protocol $24 \to 8$, batch size 64, AdamW, learning rate 10^{-3} with cosine annealing, gradient clipping 1.0, $\leqslant$ 200 epochs, validation-based checkpoint selection, five runs).

Model	Configuration
PI-DLinear	series-decomposition kernel 25; temporal predictors shared across channels (`individual=False`, two 24×8 matrices); learnable interaction operator with cohesion/alignment weights; 402 trainable parameters
DLinear	identical to PI-DLinear without the physics-informed operator; 400 parameters
Transformer	input projection $D \to 256$ with learned positional embedding; 2 encoder + 2 decoder layers, 4 heads, feed-forward dimension 1024
TCN	1×1 input projection; 2 dilated causal residual blocks (kernel 3, dilations 6 and 12, 32 filters, weight normalization); output head $32 \to 128 \to 8D$
SRU	2 SRU layers, hidden size 256, dropout 0.1
Autoformer	official architecture; $d_{\text{model}} = 128$, 2 heads, 2 encoder and 2 decoder layers, $d_{\text{ff}} = 128$, moving average 7, factor 5, GELU, value embedding without time features
iTransformer	variate tokens; embedding $24 \to 256$; 2 encoder layers, 2 heads, feed-forward dimension 256, dropout 0.1; projection $256 \to 8$
LaGNA	two-stage gray-box protocol: (1) message-passing GNN with hidden width 100 (cohesion kernel $c(r)$, alignment kernel $a(r)$, self-propulsion $F(\lvert v\rvert^2)$; diffusion term trained via NLL, not used at prediction); 10 epochs $\times$ 200 steps per epoch, batch 64, Adam with learning rate 10^{-4}, weight decay 10^{-8}; perception radius = 0.6-quantile of training pairwise distances; (2) sparse symbolic regression with an elementary-function library and a polynomial residual basis of degree 8, at most 8 terms; prediction by drift-only integration of the discovered equations over the test segment, sliced into the same $24 \to 8$ windows

Table D2 Trainable parameter counts on the three real datasets. N is the number of agents; the flattened input dimension is $6N$ for the 3D datasets (mobbing, transit) and $4N$ for the 2D fish dataset. The parameter counts of PI-DLinear, DLinear, and iTransformer do not depend on N. Autoformer's count follows its official architecture under the configuration of Table D1 (≈). LaGNA's GNN stage operates per agent and is dimension-specific (3D for the bird datasets, 2D for fish).

Model	Mobbing (N=53)	Transit (N=8)	Fish (N=10)
Transformer	4,106,814	3,968,304	3,964,200
TCN	355,152	67,872	59,360
SRU	1,327,600	703,360	684,864
DLinear	400	400	400
Autoformer	≈816,702	≈574,512	≈567,336
iTransformer	800,008	800,008	800,008
LaGNA	123,306	123,306	123,406
PI-DLinear	**402**	**402**	**402**